TURING 图灵程序设计丛书

高并发×低延迟 如何并存?

游戏开发者奥斯卡 CEDEC AWARDS 2011 最优秀著作奖

网络游戏
核心技术与实战

MMORPG / 动作类游戏 / 多人游戏 / 非同步I/O / 套接字
RPC / 事件驱动 / 实时 / 在内存中 / 多核 / 并行处理 / 吞吐量 / 自动化测试
空间分配 / 积分管理 / 支付 / 基础设施 / 开发体制

【日】中嶋 谦互 著
毛姝雯 田剑 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

网络游戏
核心技术与实战

【日】中嶋 谦互 著
毛姝雯 田剑 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

网络游戏核心技术与实战 /（日）中嶋谦互著；毛姝雯，田剑译. --北京：人民邮电出版社，2014.4
（图灵程序设计丛书）
ISBN 978-7-115-34935-4

Ⅰ. ①网… Ⅱ. ①中… ②毛… ③田… Ⅲ. ①游戏程序—程序设计 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字（2014）第043586号

内 容 提 要

本书从游戏策划与编程、系统架构、服务器运维、开发团队管理等方面全景展现网络游戏核心技术。作者使用大量图表，生动翔实地描述了网络游戏的特点和架构，并以C/S MMO游戏和P2P MO游戏为例，通过实际代码告诉开发者如何应对实时、大数据量通信的挑战，在不使用昂贵的中间件的基础上，从零开始实现趣味性强的多人网络游戏系统。此外，本书还从游戏运营和基础设施建设等角度，向读者展现了网络游戏技术的全貌。

本书适合作为综合性的网络游戏开发的参考书籍，无论是专业游戏开发技术人员还是游戏制作人、运营者都可以从中获得启发与收获。

◆ 著　[日]中嶋谦互
译　毛姝雯　田　剑
责任编辑　徐　骞
责任印制　焦志炜
◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京天宇星印刷厂印刷
◆ 开本：800×1000　1/16
印张：29
字数：680千字　2014年4月第1版
印数：1-4 000册　2014年4月北京第1次印刷
著作权合同登记号　图字：01-2013-0796号

定价：99.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

版权声明

关于本书

2011 年 1 月，苹果公司 App Store 的应用程序下载量超过了 100 亿次[①]。此外，谷歌公司的 Android Market、Chrome Web Store 以及 Amazon Appstore（亚马逊公司的 Android 应用商店，2011 年 3 月 22 日上线）、Facebook Credits 虚拟货币系统、游戏领域的网络游戏平台维尔福软件公司的 Steam 平台等，无论是面向通用操作系统的应用商店还是支付系统、游戏通信机制等都大量涌现，整个行业迎来了爆发式增长。

在每个应用商店，下载量最大的分类都是游戏，占到了总下载量的一半以上。Facebook 平台上甚至绝大多数销售额都由游戏贡献。现在，特别是智能手机和 PC 平台、画面精美的 3D 游戏，以及有玩法多样的对战游戏一般都在 10 美元以下，已经非常便宜了。再过几年，那些（开发成本）几百亿到上千亿日元的游戏应该也能很容易下载到。游戏是一种非常廉价的娱乐方式，可以满足人们放松心情这一基本需求，是一种主流的娱乐活动。

几乎所有的游戏都在增加联网的功能，让游戏向更实时、画面更加精美、利用更大的数据库、算法更加智能的方向发展。

随着市场规模的扩大，各种游戏的开发方式受到了更多的关注。虽说都是游戏，但将纸团扔进垃圾桶的 iPhone 游戏 *Paper Toss* 和即时通信大规模在线用户《魔兽世界》，在开发技术上是截然不同的。如果将它们比作交通工具，则一个是自行车，一个是新干线。它们在开发成本和销售额上有几万倍的差距，需要的技术也完全不同。只有拥有大量人才和资金的企业才有能力开发并且运营大型实时网络游戏。但如果换成 Web 服务的话，即使承载了数十万的用户，开发者一人也能够胜任开发和维护工作。两者之所以有这样的差别，我想其中一个原因就是开发网络游戏所需要的知识并没有得到充分的共享。

本书以实时通信、大数据量通信的多人网络游戏开发为中心，详细介绍了普通开发者如何在不使用昂贵的中间件或者特殊开发环境的基础上，独自从零开始实现有趣的多人网络游戏系统，并讲解了 C/S MMO 游戏和 P2P MO 游戏这两个典型的开发案例。此外还从游戏运营和基础设施架构等角度，向读者展现了支持网络游戏技术的全貌。本书的内容主要面向游戏开发技术人员，但无论是对游戏制作人还是运营者，本书都非常具有参考价值。

希望不仅仅是大企业，今后更多的独立开发者或者小企业也能够开发网络游戏，为玩家提供更多的充满创意的产品。

2011 年 2 月　中嶋谦互

网络游戏的开发技术每天都在不断进步。
如果对本书有什么疑问或者建议可以通过 Twitter 和 @ringo 联系，欢迎提问。

① 2013 年的最新统计，App Store 应用累计下载次数超过 600 亿。——译者注

本书结构

本书的结构如下。要理解书中与编程相关的章节（第 0 章、第 4 章、第 5 章），需要具备一定的 C 或者 Java 等一般编程语言基础知识。其他章节（第 1～3 章、第 6～8 章）的内容不需要编程知识也可以理解。

第 0 章　[快速入门] 网络游戏编程

网络和游戏编程的技术基础

本章详细介绍专业的网络游戏开发者和其他行业的开发者（Web 开发者或者企业应用开发者等）在技术上的不同点。

第 1 章　网络游戏的历史和演化

游戏进入了网络世界

第 2 章　网络游戏到底是什么

网络游戏面面观

第 3 章　网络游戏的架构

挑战游戏的可玩性和技术限制

总结了网络游戏的历史、背景、商业模式、不同分类的区别以及架构等背景知识。

第 4 章　[实践]CS MMO 游戏开发

长期运行的游戏服务器

第 5 章　[实践]P2P MO 游戏开发

没有专用服务器的动作类游戏的实现

介绍了可以涵盖大部分网络游戏的两种实现方式，通过案例游戏和实际代码带领读者体验开发流程。

第 6 章　网络游戏的辅助系统

完善游戏服务的必要机制

第 7 章　支持网络游戏运营的基础设施

构筑、负荷测试、开始运营

第 8 章　网络游戏的开发体制

团队管理的挑战

介绍了对于网络游戏非常重要的运营相关的辅助技术和思维方式。

关于本书的结构：

同类书籍一般会首先针对网络游戏这个主题介绍相关的定义，说明背景，提出课题，简要地说明解决方法，然后通过案例程序具体介绍开发技术。本书也基本上沿用了这个方式，但是在书的开始部分增加了第 0 章，用来简要介绍开发网络游戏时的网络编程和游戏编程知识，方便读者快速学习。

第 0 章尽可能涵盖了全书的专业词汇，只需要快速浏览即可了解实际开发中需要用到的技术。如果在接下来的章节遇到了不太理解的内容，也可以回到第 0 章查阅以加深理解。

如果读者对第 0 章介绍的技术都非常熟悉，那一定可以在任何一家网络游戏开发公司成为核心开发人员。如果对这些知识都不太了解也不用担心，第 0 章之后会详细解释为什么需要用到这些技术，可以反复查阅相关内容进行学习，这样一定可以加深理解。

案例游戏和视频录像

由于篇幅所限，书中所附案例采用了一部分伪代码。同时我们为大家准备了可以实际运行的案例游戏（包括 C/S MMO 和 P2P MO 两种类型）作为参考。下载地址请参见稍后的本书附加信息。源代码仅仅是为了便于读者理解编程方法，请作为游戏运行情况和书中代码案例的参考。

除此之外，我们还为大家准备了以上两种案例游戏实际运行时的视频录像，请参见稍后的本书附加信息。

案例游戏的运行环境

为了正常运行本书的案例游戏，需要确保具备以下软件环境。使用时请确认拥有合适的软件授权。

- Mac OS X v10.6（Snow Leopard）
- Xcode　http://www.apple.com/jp/macosx/developers/
- Boost 1.41.0　http://www.boost.org/LICENSE_1_0.txt
- SDL 1.2.14　http://www.libsdl.org/license-lgpl.php
- **通信中间件 VCE ※ 请参考第 4 章相关章节**
- MySQL 等其他软件

※ 详细清单请参考压缩包中的 readme.txt 文件。

关于通信中间件 VCE

史克威尔艾尼克斯公司拥有通信中间件 VCE 的著作权和知识产权。因为本书旨在提高行业的技术水平，所以在本书规定的范围内（包含附属规定）可以授权使用该软件。

购买本书的读者，可以访问稍后的本书附加信息中的 URL，接受技术评论社网站的附属规定后可以得到授权使用 VCE（技术评论社授权版）。

另外，本书案例游戏的压缩包中所含的 VCE 程序为试用版，在启动最多一小时后会无法继续通信。史克威尔艾尼克斯公司、原书作者、技术评论社、人民邮电出版社以及译者不承担任何因为使用 VCE 所造成的损失。

使用须知 ※ 请仔细阅读 ※

- **案例游戏的压缩包中附带的数据和程序只是作为本书正文相关内容的参考和辅助材料。请不要用作其他目的。**
- **史克威尔艾尼克斯公司、程序或者数据的作者、开发者 / 提供者、原书作者、技术评论社、人民邮电出版社以及译者不承担任何因为使用本书案例游戏程序或数据所造成的损失。请读者自行承担相应责任。**

- **使用关于本书所收录的软件、程序前请注意，对于相关软件的信息和使用方法，我们不接受包括电话在内的任何方式的技术支持。**
- **书中的案例游戏的程序代码是完全开源的，可以不经任何许可自由使用。**
- **其他的注意事项请参考案例游戏压缩包内的 readme.txt 文件。**

本书附加信息

本书的附加信息包含了案例游戏的下载地址和视频下载地址，如下所示。

- **案例游戏的下载地址和本书的主页：**

  ➡ http://gihyo.jp/book/2011/978-4-7741-4580-8/support

- **游戏视频和特别内容下载地址：**

  ➡ http://gihyo.jp/magazine/wdpress/plus/978-4-7741-4580-8

※ 请注意，上述下载地址可能会有突然中断访问的情况。

致谢

本书内容基于笔者到目前为止从参与的网络游戏开发项目中获得的经验。项目很多，在此就不一一列举了，我之所以获得如此多宝贵的经验，离不开各个项目组的同仁们的帮助。

此外还要感谢史克威尔艾尼克斯公司秉承提高业界技术水平的宗旨，为本书无偿提供通信中间件 VCE。

最后，在本书写作的近两年的时间中，我放弃了新年假期，减少了陪伴孩子的时间，给家里增添了许多负担。特别感谢妻子能够给与充分的理解和支持。

专业术语

本书涉及的内容比较广，使用了很多专业术语。下面整理了本书中出现的网络 / 游戏 / 编程相关的专业术语作为辅助资料。因使用场合和上下文的不同，每个术语的意思可能会发生变化，笔者基于本书内容对以下术语进行说明，供大家参考。书中的网络环境是指互联网，服务器操作系统为 Linux。

16 毫秒 / 帧速率（Frame Rate）

电子游戏使用的光栅显示器是普通电视时，图像一般每秒更新 60 次。图像更新的时间叫做帧，1 秒 60 次即 1 次 16 毫秒（0.0167 秒 =16.7 毫秒）。16 毫秒是玩家可以识别的游戏画面改变的最短时间间隔。

ARPG（Action Role Playing Game）

角色扮演类游戏中动作性较强的实时游戏，也指包含冒险游戏特征的游戏。

bot

外挂。模拟游戏玩家自动访问游戏服务器、高效率地进行游戏、积累分数以及进行恶意的经济欺诈的程序。测试外挂是指开发者准备的用来自动化测试的客户端程序。

CPU 周期（CPU Cycle）

CPU 处理操作的最小单位。1GHz 的 CPU 一秒有 10 亿个 CPU 周期，以执行的命令数而言，1 秒可以执行 10 亿次。

根据命令类型的不同，执行需要的 CPU 周期少则不到 1 个周期，多则有几百个。

FPS（First-Person Shooter）

第一人称射击游戏。

I/O（Input/Output）

输入 / 输出。包括网络 I/O、磁盘 I/O 等。服务器程序的 I/O 基本都是网络 I/O。

MMO（Massively Multiplayer Online）

大型多人网络游戏，本书是指有大规模用户、多人在线的网络游戏。也叫 MMOG。

MO（Multiplayer Online）

多人网络游戏，本书是指参与用户相对较少的网络游戏。也叫 MOG。

RPC（Remote Procedure Call）

远程过程调用，是指调用其他计算机的处理。例如，当客户端需要命令服务器做某个处理然后得到返回结果时会使用该技术。

RPG（Role Playing Game）

角色扮演游戏，根据游戏背景设定，由玩家扮演特定角色的游戏。

TCP（Transport Control Protocol）

传输控制协议，支撑整个互联网的可靠数据通信协议。可以根据需要续传 IP 数据包，确保大的数据可以正常传输。但是，在连接速度较慢时，为了提高传输效率需要占用大量的内存。

World of Warcraft（《魔兽世界》）

暴雪公司过去 5 年全世界市场占有率第一的 MMORPG。总注册用户 1200 万，仅在中国，同时在线用户就达 100 万。

并行（Parallel）

包括物理上的多个处理同时进行，以及时间上的并发（Concurrent）处理。就像 CPU 中的命令和任务之间的区别。通过并行处理提高速度比较困难，所以基本方针是充分考虑处理器计算能力，通过在策划层次进行讨论，或者在算法上下功夫减少计算量。

部署（Deploy）

是指部署应用程序。服务器部署是指将最新版服务器程序安装到各个服务器上来更新版本的相关操作。

持久性（Persistent）

在数据库中，持久性是指需要持久化的时间长度，包括游戏玩法中必要的时间和游戏进行所需时间。竞速游戏的数据一般只需要保持几分钟，之后就可以丢弃，所以持久性较低，需要保存的数据量也比较小。但是 MMORPG 等不断进行的游戏需要较高持久性，数据量也比较大。根据持久性需求的不同，数据应该以什么形式、用什么物理介质来保存也会有所区别。

带宽（Bandwidth）

是指网络游戏开发中，通过网络传输数据的传输速率。也叫带宽幅度。

多进程编程（Multi-Process Programming）

灵活使用多个进程的编程方式。同时运行多个进程可以有效利用多核 CPU 的处理能力。

辅助系统（Additional System）

相对于游戏主体内容以外的辅助功能系统，例如玩家匹配、玩家成绩管理（积分管理）、排名以及通信功能等。多数情况下可以使用第三方的程序库或者服务。

负荷（Load）

是指 CPU 或者网络等承载的工作量。例如，处理复杂计算时 CPU 的负荷比较高。发送和接收大量数据时网络的负荷较高。许多场合都可以使用，例如 CPU 负荷、I/O 负荷以及服务器负荷等。

负载均衡（Load Balancing）

是指分散负荷。例如将一台数据库承担的负荷分散到多台数据库。

共享内存（Shared Memory）

是指在多个进程间共享内存数据。例如共享运动物体的坐标、种类以及运动方向等信息。

缓存（Cache）

为了高速读取数据而把数据暂时放在特殊区域。例如，磁盘访问比较慢时，可以把文件内容放在（缓存在）内存中，这样就可以高速读取数据。该机制被广泛应用在 CPU 缓存、缓存内存、浏览器缓存以及缓存服务器等地方。

进程（Process）

进程是指操作系统上运行的程序的实体，和其他程序相分离，独立运行。进程与进程之间可以访问的资源（内存、Socket 等）也是分离的。

进程间通信（Inter-Process Communication）

在多个进程间通信。是指多个进程间传送数据或共享数据的技术。

竞态条件（Race Condition）

是指同一个资源（内存地址等）被两个以上的使用者访问时发生的程序状态。会引起死锁（Dead Lock，互相等待对方处理结果的情况）等问题。

扩展性（Scalable）

是指可以扩展系统性能。在网络游戏中需要应对用户的增长和饱和，所以希望性能和功能可以轻松扩展。

浏览器（Browser）

浏览软件。网络游戏中的游戏浏览器范围较广，泛指将服务器网站管理的游戏进度信息展示给玩家的软件。例如使用 C++ 语言开发的面向 3D 游戏的专用程序，或者 Flash 游戏使用的 Google Chrome 等 Web 浏览器。和一般浏览 Web 服务器数据的 Web 浏览器有所区别。请参考“游戏客户端”的解析。

轮询（Polling）

定期询问数据是否送达或者是否接收到的机制。太过频繁的轮询会无端增加 CPU 的负荷。

瓶颈（Bottleneck）

系统中性能最弱的部分。系统的其他部分即使再快，如果有一个地方（瓶颈）处理比较慢，就会影响整体的性能。

冗余（Redundancy）

是指作为预备而重复配置。游戏数据的冗余是指将数据在不同地方重复保存（主数据和备份数据的关系）。

事件驱动（Event Driven）

在事件发生时进行处理的编程方式。事件的类型包括接收到数据、鼠标移动等。事件驱动的编程方式常用在网络开发和游戏开发中。

数据包（Packet）

数据的传输单位。数据包通信是指将数据分割并添加控制信息后发送、接收后再合并的通信方式。TCP 协议的数据通信单位是数据段（Segment），UDP/IP 协议为数据报（Datagram）。网络游戏开发会经常面临数据包延迟的挑战。

数据中心（Data Center）

安置提供服务的服务器设施。配备了维持服务器所必需的电源、空调和防灾设施。

套接字 API（Socket API）

处理网络文件描述符中的套接字的 API。个别的函数 / 系统调用（socket、connect、accept）相关内容可以参考第 0 章。

同时连接数（Number of Simultaneous Connections）

可以同时连接服务的用户数。游戏策划阶段一般指最大的同时在线数。运营阶段是指同时在线用户的瞬间值。和网络连接中（通信线路）的连接数有所区别。

图元（Sprite）

是指电子游戏中使用的可以高速显示的小图像。事先准备好玩家角色移动等状态的小图片，可以通过指定图像位置，在画面的任意位置显示角色。

吞吐量（Throughput）

是指系统在一定时间内处理的数量。例如，1 秒处理 1000 次的系统就比只能处理 100 次的系统的吞吐量高。

网络拓扑（Network Topology）

是指网络中所含的各个计算机以什么结构相连接。计算机是节点，连接叫做边界。包括星状结构、总线结构和网状结构等，可以帮助分析和设计网络结构。

文件描述符（File Descripter）

在 Unix 系的操作系统中，除了文件以外，网络、块设备等操作系统管理的输入输出资源也采用了文件形式。文件描述符是一个整数（C 语言中的 int 型），操作系统会通过程序分配文件描述符，并使用它来控制数据的输入输出。通过文件描述符，可以让网络的输入输出像文件的输入输出一样进行。

线程（Thread）

比进程更细分的程序执行单位。共享资源比较多的情况下，线程间的协调比进程更加容易。只使用单一线程执行处理的方法叫做单线程（Single Threading），使用多线程同时进行处理的方法叫做多线程（Multi-Threading）。

延迟（Latency）

处理所需要的时间。例如，如果玩家按下按键到画面反应的时间间隔为 1 秒，那这一延迟就过长了。场合不同，单位也不一样，CPU 访问的时间间隔为纳秒、内存访问是几十到几百纳秒、SSD 访问是几百微秒、硬盘访问是几十毫秒、网络延迟是毫秒到秒。

游戏客户端（Game Client）

是指在玩家的 PC 或者游戏机等机器上安装的，启动后可以显示游戏画面、接受用户输入输出的软件。例如，需要高性能绘图和输入输出功能的 ARPG 或者 3D 画面的 MMORPG 等网络游戏都需要开发专用的客户端。游戏客户端也可以称为客户端软件。请参考“浏览器”的说明。

游戏逻辑（Game Logic）

相当于 Web 应用的业务逻辑，是指连接游戏进度信息和用户界面信息的算法。

云（Cloud）

在云计算（Cloud Computing）中主要是指服务器端的计算机群。在单纯的主机托管中，包括存储、负载均衡、付费系统、日志解析等服务器架构中的计算机资源可以根据需要即时调整。尽管非常方便，但是需要将重要的保密信息和用户资料交给云服务提供商。

在内存中（on Memory）

是指把数据放在内存中，可以在几个 CPU 时钟周期，也就是几纳秒到几百纳秒之间获取到数据的状态。

中间件（Middleware）

将应用程序普遍使用的功能进行集成的专业化软件。比如将通信处理等对性能和适应性要求较高的处理做成程序库，并将复杂的处理隐藏起来，通过访问 API 即可完成复杂的工作。

纵向扩展 / 横向扩展（Scale-up/Scale-out）

纵向扩展是指增加内存、升级 CPU 等，通过提升单台服务器的性能来改善系统性能的方法。横向扩展是指通过增加服务器台数来提供系统性能的方法。单台服务器有性能瓶颈，所以大型网络游戏需要具备可以通过增加服务器数量来横向扩展的性能。

阻塞 / 非阻塞（Blocking / Non-Blocking）

阻塞是指处理完成之前持续等待。例如，收到数据前持续等待的程序（阻塞程序），在等待期间不能进行其他处理。采用非阻塞（不持续等待）处理可以解决这个问题。也可以叫做同步调用和非同步调用。

目录